Digital Control: Part 2

ENGI 7825: Control Systems II

Andrew Vardy

Mapping the s-plane onto the z-plane

- We're almost ready to design a controller for a DT system, however we will have to consider where we would like to position the poles
- We generally understand how to position desirable poles in the s-plane
  - Although this does remains somewhat of a "black art" as there are various arbitrary choices and rules-of-thumb at play
- If we understand how to position poles in the z-plane we can do direct digital design. Alternatively, we can position poles in the s-plane and then find out where they lie in the z-plane.

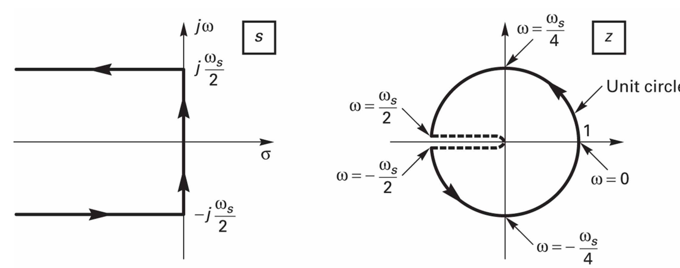
- We have already seen that poles in the s-plane and z-plane are related by

$$z = e^{sT}$$

- We'll consider particular mappings from parts of the s-plane. We have already seen that the j! axis corresponds to the unit circle in the z-plane. In the following, $s = \sigma + j\omega$ and $\omega = 0$.

$$z = e^{sT} = e^{\sigma T} e^{j\omega T} = e^{j\omega T} = 1\underline{/\omega T}$$

- Fundamentally, there is a limitation on the signal frequency that can be represented by the z-transform. That limit is $\omega = \omega_s / 2$ where $\omega_s = 2\pi / T$.

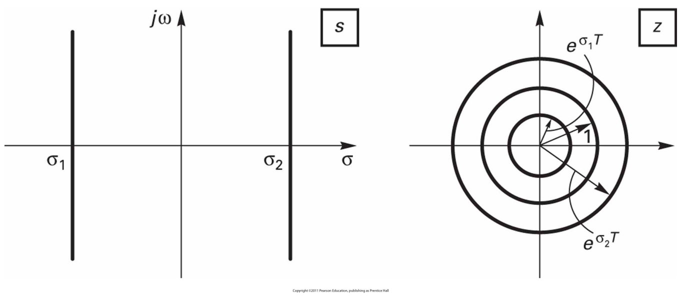


- That portion of the j! axis which lies in the range $[-j\omega_s/2, j\omega_s/2]$ maps onto the unit circle.
- So poles on the unit circle in the z-plane correspond to pure sinusoids and therefore signify a marginally stable system.
- As we have already seen, poles inside the unit circle correspond to exponentially decaying sinusoids. If all poles lie within the unit circle then we have asymptotic stability. Poles outside the unit circle correspond to exponentially growing sinusoids, and therefore instability.

For s = σ + jω if σ is held constant (lets say we set it to a value of σ_1) and ω is allowed to vary we get

$$z = e^{\sigma_1 T} e^{j\omega T} = e^{\sigma_1 T} \angle \omega T$$

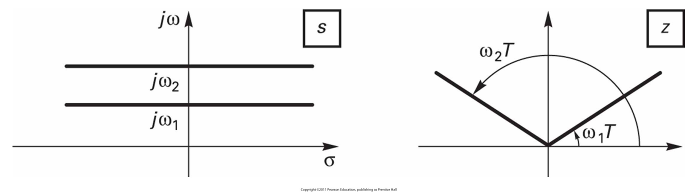
This corresponds to vertical lines in the s-plane and circles in the z-plane (including the unit circle).



What if we do the opposite? That is, for s = σ + jω we hold ω constant (at ω_1) if σ is allowed to vary allowed to vary we get

$$z = e^{\sigma T} e^{j\omega_1 T} = e^{\sigma T} \angle(\omega_1 T)$$

This corresponds to horizontal lines in the s-plane and rays emanating from the origin in the z-plane.



Lets consider pairs of poles located at s = σ ± jω. We know that such a pole pair corresponds to a term of the form $ke^{\sigma t}\cos(\omega t + \psi)$. We can also define this pair of poles in polar coordinates as (r, ±θ) as below:

$$z = e^{sT}\big|_{s=\sigma \pm j\omega} = e^{\sigma T} e^{\pm j\omega T} = e^{\sigma T} \angle \pm\omega T = r \angle \pm\theta$$

In particular we would like to position the poles of a second-order system which have the following locations:

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

Now translate to the z-plane:

$$z_{1,2} = e^{sT}\big|_{s=s_{1,2}} = e^{-\zeta\omega_n T} e^{\pm j\omega_n T\sqrt{1-\zeta^2}} = re^{\pm j\theta}$$

$$\text{where } r = e^{-\zeta\omega_n T} \text{ and } \theta = \omega_n T\sqrt{1-\zeta^2}$$

We can then solve for the relationship between (r, ±θ) and (ζ, ω_n):

$$\zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} \qquad \omega_n = \frac{1}{T}\sqrt{\ln^2 r + \theta^2}$$

The relationships between z-plane pole locations and (ζ, ω_n) is somewhat complex, geometrically:

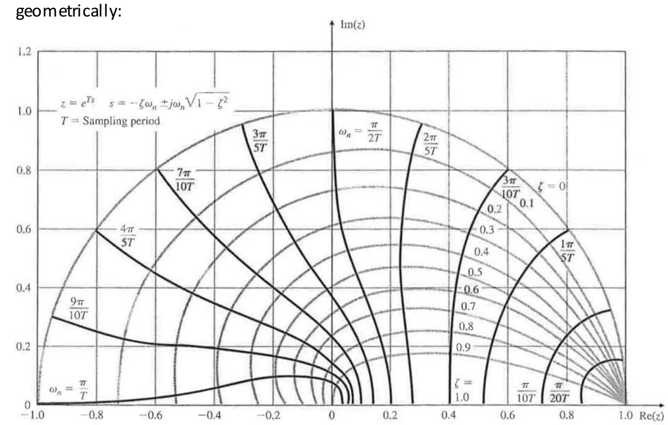


Figure 8.4
Natural frequency (solid color) and damping loci (light color) in the z-plane; the portion below the Re(z)-axis (not shown) is the mirror image of the upper half shown

$$\zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} \qquad \omega_n = \frac{1}{T}\sqrt{\ln^2 r + \theta^2}$$

These relationships between the locations of a pole pair at (r, ±θ) in the z-plane and second order system parameters (ζ, ω_n) allow us then to relate pole locations to "boss parameters" such as %OS and settling time.

Example:

We have a DT system with the following closed-loop characteristic polynomial:

$$z^2 - z + 0.632 = (z - 0.5 - j(0.618)(z - 0.5 + j0.618) = 0$$

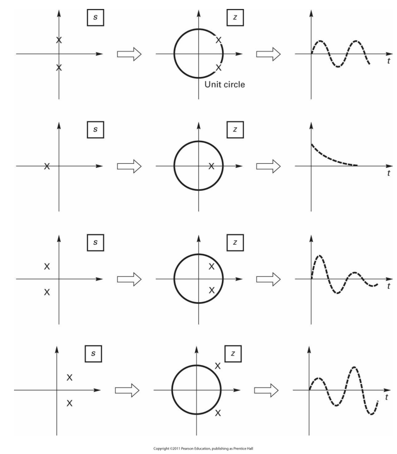
Get the pole locations in the z-plane in terms of (r, ±θ) then obtain the 2nd order parameters (in this example T = 1s which is rather slow):

$$z_{1,2} = 0.5 \pm j0.618 = 0.795\angle \pm 0.890(\text{rad}) = r\angle\pm\theta$$

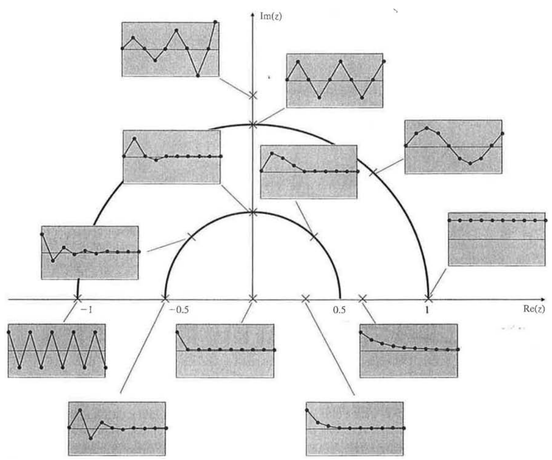
$$\zeta = \frac{-\ln(0.795)}{\sqrt{\ln^2(0.795) + (0.890)^2}} = 0.250$$

$$\omega_n = \frac{1}{1}\sqrt{\ln^2(0.795) + (0.890)^2} = 0.919$$

The examples below illustrate 4 different configurations of s-plane and corresponding z-plane pole locations and the resulting signals produced.

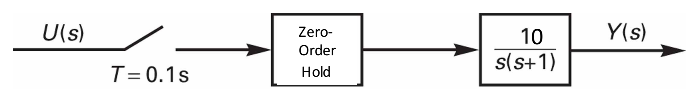


The following plot from Franklin gives a similar picture:



Digital State Feedback Design

- State feedback can be applied to sampled data systems in almost exactly the same way as for CT systems
  - The only real difference is that we place eigenvalues in the z-plane, not the s-plane
- We proceed by example. Assume we have the following servomotor system (again):



In the previous set of notes we developed the following discretized state-space model for this system:

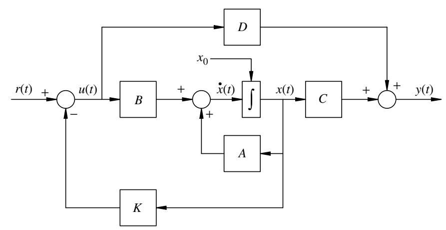
$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} m(k)$$

$$y(k) = [1 \quad 0]\,\mathbf{x}(k)$$

$x_1(k)$ represents the angle of the motor shaft (measureable by encoder count).
$x_2(k)$ represents the shaft speed (measureable by a tachometer, rate gyro, or by rate of encoder counts).

It is important to consider whether the state variables are measureable because otherwise full-state feedback cannot be applied.

Here is our usual picture of a state feedback controller:



This example differs in that it has been discretized, but also in that the goal is to set the motor's shaft angle to zero. That makes this controller a **regulator**. A regulator is a controller or compensator that works to move one or all state variables to zero. So we can say there is no r(t), or equivalently that r(t) = 0.

In regulator design (for n=2) the input to the plant is defined as

$$u(k) = -K_1x_1(k) - K_2x_2(k) = -\mathbf{Kx}(k)$$

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} m(k)$$

$$y(k) = [1 \quad 0]\,\mathbf{x}(k)$$

Problem specification: Reduce settling time to 4 seconds. (Nothing else is mentioned which means we don't particularly care about other specifications such as %OS).

Start by looking at the open-loop system and its characteristics. We will need the current characteristic polynomial (computed as usual except that we use |zI – A| instead of |sI – A|).

$$a(z) = |zI - A| = (z-1)(z-0.905) = z^2 - 1.905z + 0.905$$

The design process that follows goes from a unity feedback system (which is identical to state feedback with K_1 = 1, K_2 = 0). That unity feedback system has the following characteristic polynomial:

$$a_{uf}(z) = z^2 - 1.9z + 0.91$$

The eigenvalues of the unity feedback system can be obtained from the quadratic formula then converted to polar form:

$$z_{1,2} = 0.954\,\angle \pm 0.091\,\text{rad} = r\angle \pm \theta$$

$$z_{1,2} = 0.954\,\angle \pm 0.091\,\text{rad} = r\angle \pm \theta$$

Work out the second-order parameters:

$$\zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} = \frac{-\ln(0.954)}{\sqrt{\ln^2(0.954) + (0.091)^2}} = 0.46$$

$$\omega_n = \frac{1}{T}\sqrt{\ln^2 r + \theta^2} = 1.0246$$

Current settling time:

$$T_s = \frac{4}{\zeta\omega_n} = 8.47$$

Since we don't care about %OS lets just change ω_n. To bring the desired settling time down to 4 seconds we modify ω_n and then get the desired pole locations:

$$\omega_n' = \frac{4}{\zeta 4} = 2.17$$

$$z_{1,2} = e^{sT}\big|_{s=s_{1,2}} = e^{-\zeta\omega_n T}e^{\pm j\omega_n T\sqrt{1-\zeta^2}} = re^{\pm j\theta}$$

$$\lambda_{1,2} = 0.905\angle \pm 11.04^\circ = 0.888 \pm j0.173$$

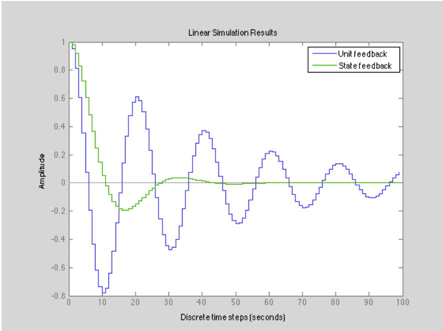
$$\lambda_{1,2} = 0.905\angle \pm 11.04^\circ = 0.888 \pm j0.173$$

Now we can get the desired characteristic polynomial:

$$(z - 0.888 - j0.173)(z - 0.888 + j0.173) = z^2 - 1.776z + 0.819$$

We continue to design the K gain vector in the usual way. The system is not in CCF so we use Bass-Gura and obtain K = [0.445 0.113].

The following shows the resulting improvement in system response ($x(0) = [1\ \ 0]^T$).



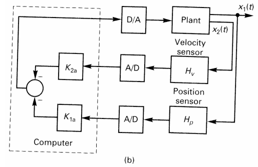


FIGURE 14.3
Hardware implementation for the design of Example 14.1.